UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/745,104	12/20/2000	Ryo Inoue	10559-391001/P10256-AD1	6933

20985 7590 11/19/2004		EXAMINER
FISH & RICHARDSON, PC		HARKNESS, CHARLES A
12390 EL CAMINO REAL	ART UNIT	PAPER NUMBER
SAN DIEGO, CA 92130-2081	2183	

DATE MAILED: 11/19/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

-- *The MAILING DATE of this communication appears on the cover sheet with the correspondence address --*

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.
- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on _02 September 2004_.

2a)☐ This action is **FINAL**.   2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) _1,2 and 4-30_ is/are pending in the application.

    4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) _1-2, and 4-30_ is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

    Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

    Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

    a)☐ All   b)☐ Some *   c)☐ None of:

      1.☐ Certified copies of the priority documents have been received.

      2.☐ Certified copies of the priority documents have been received in Application No. _____.

      3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

    * See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☐ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
    Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
    Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application (PTO-152)

6)☐ Other: _____.

DETAILED ACTION

Claim Rejections - 35 USC § 102

1.   The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the

basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

2.   Claims 1-18 and 20-25 are rejected under 35 U.S.C. 102(b) as being anticipated by

Atkins et al., U.S. Patent Number 5,898,866 (herein referred to as Atkins).

3.   Referring to claim 1 Atkins has taught a method comprising:

defining, within a computer program, loop conditions for a first hardware loop (Atkins

column 2 lines 14-50; the conditions are in the SETLOOP instruction, which is software being

run as part of the program);

first propagating a first of said loop conditions of said first hardware loop via a first

pipeline of a pipelined processor (Atkins figure 2 reference 20A, column 2 lines 14-50; shows

that Atkins has taught a pipelined processor); and

second propagating in parallel with said first propagating a second of said loop conditions

for said first hardware loop (Atkins column 4 lines 12-44, column 3 lines 1-20; Atkins has

several pipelines, where several different loops can be executed in parallel with each other, since

more than one SETLOOP instruction could be present and each one can be present in a different

pipeline; also the situation would exist where the two pipelines are executing different instances

of the same instruction group, or same SETLOOP instruction, thereby being the same hardware

loop e) via a second pipeline of the pipelined processor (Atkins figure 2 reference 20B, column 2 lines 14-50; shows that Atkins has taught a pipelined processor).

4.   Referring to claim 2 Atkins has taught further comprising:

writing the loop conditions for said first hardware loop to a first set of registers prior to propagating the loop conditions for said first hardware loop (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C), and using said registers to begin calculating parameters based on said loop conditions for said first hardware loop prior to said propagating (Atkins figure 4; the registers included in the loop hardware shown in figure 4, are used to calculate the current parameters for the loop, before propagating the loop conditions again, i.e. the counts are decremented before the new values are propagated to determine if the process is at the end of the loop, and if it is, the TOP register 50 is then propagated to start the next loop; the 'propagating' is not clearly defined and can be broadly interpreted to mean propagating any of conditions at any time in the process), and

writing the loop conditions to a second set of registers after propagating the loop conditions (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C).

5.   Referring to claim 4 Atkins has taught further comprising propagating a third loop of said conditions via a third pipeline (Atkins figure 2 reference 20C, column 2 lines 14-50; shows that Atkins has taught a pipelined processor).

6.   Referring to claim 5 Atkins has taught further comprising generating the loop conditions

of the hardware loop prior to writing the loop conditions to the first set of registers (Atkins

column 2 lines 14-49; the conditions would have to be generated before being stored in a

register, otherwise the hardware would not know the value to store).

7.   Referring to claim 6 Atkins has taught wherein generating the loop conditions comprise

calculating at least one of the loop conditions from program counter relative data in a loop setup

instruction (Atkins column 10 lines 16-24; the value of the program counter after it has been

incremented is stored as the top of the loop value).

8.   Referring to claim 7 Atkins has taught a method comprising:

first calculating a first loop condition of a first hardware loop from a loop setup

instruction using a first arithmetic logic unit in a first pipeline (Atkins figures 2-4, column 2 lines

14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is

present in all of the execution units, 20A-C); and

second calculating a second loop condition of said first hardware loop from the loop

setup instruction using a second arithmetic logic unit in a second pipeline (Atkins figures 2-4,

column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the

setloop, which is present in all of the execution units, 20A-C);

using results of said first calculating and said second calculating for propagating loop

information to hardware registers associated with calculating parameters of said first hardware

loop; and beginning to calculate said parameters using said first hardware loop, based on said

loop conditions, prior to said propagating (Atkins figure 4; the registers included in the loop

hardware shown in figure 4, are used to calculate the current parameters for the loop, before

propagating the loop conditions again, i.e. the counts are decremented before the new values are

propagated to determine if the process is at the end of the loop, and if it is, the TOP register 50 is

then propagated to start the next loop; the 'propagating' is not clearly defined and can be broadly

interpreted to mean propagating any of conditions at any time in the process; also the situation

would exist where the two pipelines are executing different instances of the same instruction

group, or same SETLOOP instruction, thereby being the same hardware loop e) via a second

pipeline of the pipelined processor).

9.   Referring to claim 8 Atkins has taught further comprising writing the first and second

loop conditions to a first set of registers (Atkins figure 2 reference 20, figure 4, column 2 lines

14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is

present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in

all of the processors is a set of 3 setloop hardware registers).

10.   Referring to claim 9 Atkins has taught further comprising:

        calculating a third loop condition of the hardware loop from the loop setup instruction

using a third arithmetic logic unit in a third pipeline (Atkins figures 2-4, column 2 lines 14-50,

column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in

all of the execution units, 20A-C); and

        writing the first, second and third loop conditions to a first set of registers (Atkins figure

2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is

the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the

registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware

registers).

11.   Referring to claim 10 Atkins has taught wherein calculating the first loop condition and calculating the second loop condition occur in parallel (Atkins column 4 lines 12-44, column 3 lines 1-20).

12.   Referring to claim 11 Atkins has taught further comprising propagating the first loop condition to a second set of registers via a first pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers for the system, the value of the TOP register is propagated to the instruction-address register set in all of the pipelines when necessary by the hardware performing a loop operation).

13.   Referring to claim 12 Atkins has taught further comprising propagating the second loop condition to the second set of registers via a second pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers for the system, the value of the TOP register is propagated to the instruction-address register set in all of the pipelines when necessary by the hardware performing a loop operation).

14.   Referring to claim 13 Atkins has taught an apparatus comprising:
a first pipeline including a first arithmetic logic unit and a second pipeline including a second arithmetic logic unit (Atkins figure 2 and figure 3 reference 20-4C), and

        a control unit coupled to the pipelines (Atkins figure 2 reference 26, column 4 lines 12-44), the control unit adapted to:

        obtain loop setup instructions for a first hardware loop from a computer program (Atkins column 2 lines 14-50; the conditions are in the SETLOOP instruction);

first calculate a first loop condition of said first hardware loop from one of said loop setup instructions using the first arithmetic logic unit in the first pipeline (Atkins figures 2-4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C); and

second calculate a second loop condition of said first hardware loop from a loop setup instruction using the second arithmetic logic unit in the second (Atkins figures 2-4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C;), pipeline in parallel with said first calculate (Atkins column 4 lines 12-44, column 3 lines 1-20; Atkins has several pipelines, where several different loops can be executed in parallel with each other, since more than one SETLOOP instruction could be present and each one can be present in a different pipeline; also the situation would exist where the two pipelines are executing different instances of the same instruction group, or same SETLOOP instruction, thereby being the same hardware loop e) via a second pipeline of the pipelined processor).

15.   Referring to claim 14 Atkins has taught further comprising a first set of registers coupled to the control unit, wherein the control unit is further adapted to write the first and second loop conditions of the hardware loop to the first set of registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers; the registers are coupled to the control unit indirectly from the execution units, since the registers are a part of the execution units and the execution units are coupled to the control unit).

16.   Referring to claim 15 Atkins has taught further comprising a third pipeline coupled to the

control unit, the third pipeline including a third arithmetic logic unit (Atkins figure 2 and figure 3

reference 20-4C), the control unit further adapted to:

calculate a third loop condition of the hardware loop from the loop setup instruction using

the third arithmetic logic unit in the third pipeline (Atkins figures 2-4, column 2 lines 14-50,

column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in

all of the execution units, 20A-C); and

write the first, second and third loop conditions of the hardware loop to the first set of

registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column

10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units,

20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3

setloop hardware registers).

17.   Referring to claim 16 Atkins has taught further comprising a second set of registers

coupled to the control unit, wherein the control unit is further adapted to propagate at least one of

the loop conditions to the second set of registers via the first pipeline (Atkins figure 4 reference

50, column 10 lines 16-24; the instruction-address registers for all of the execution units together

are a set of 3 instruction-address registers for the system, the value of the TOP register is

propagated to the instruction-address register set in all of the pipelines when necessary by the

hardware performing a loop operation).

18.   Referring to claim 17 Atkins has taught further adapted to propagate at least one of the

loop conditions to the second set of registers via the second pipeline (Atkins figure 4 reference

50, column 10 lines 16-24; the instruction-address registers for all of the execution units together

are a set of 3 instruction-address registers for the system, the value of the TOP register is

propagated to the instruction-address register set in all of the pipelines when necessary by the

hardware performing a loop operation).

19.   Referring to claim 18 Atkins has taught further comprising a second set of registers

coupled to the control unit, the control unit further adapted to:

propagate at least one of the loop conditions to the second set of registers via the first

pipeline, propagate at least one of the loop conditions to the second set of registers via the second

pipeline, and propagate at least one of the loop conditions to the second set of registers via the

third pipeline(Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address

registers for all of the execution units together are a set of 3 instruction-address registers for the

system, the value of the TOP register is propagated to the instruction-address register set in all of

the pipelines when necessary by the hardware performing a loop operation).

20.   Referring to claim 20 Atkins has taught wherein at least one of the pipelines is a data

address generation pipeline (Atkins column 6 lines 33-48, figure 3 reference D-BUS; at least a

part of the pipeline is concerned with data address generation).

21.   Referring to claim 21 Atkins has taught wherein at least one of the pipelines is a system

pipeline (Atkins figures 2-3, column 4 lines 13-44; all of the pipelines are a part of the system).

22.   Referring to claim 22 Atkins has taught an apparatus comprising a set of registers, a first

pipeline, and a second pipeline (Atkins figure 2 column 2 lines 14-50); and

a control unit coupled to the set of registers, the first pipeline and the second pipeline

(Atkins figure 2 reference 26, column 4 lines 12-44), the control unit adapted to:

first propagate at least one loop condition of a first hardware loop to the set of registers

via the first pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address

registers for all of the execution units together are a set of 3 instruction-address registers for the

system, the value of the TOP register is propagated to the instruction-address register set in all of

the pipelines when necessary by the hardware performing a loop operation); and

second propagate at least one loop condition of said first hardware loop to the set of

registers via the second pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the

instruction-address registers for all of the execution units together are a set of 3 instruction-

address registers for the system, the value of the TOP register is propagated to the instruction-

address register set in all of the pipelines when necessary by the hardware performing a loop

operation; also the situation would exist where the two pipelines are executing different instances

of the same instruction group, or same SETLOOP instruction, thereby being the same hardware

loop e) via a second pipeline of the pipelined processor); and

begin calculating using said first hardware loop, prior to completing said first and second

propagate (Atkins figure 4; the registers included in the loop hardware shown in figure 4, are

used to calculate the current parameters for the loop, before propagating the loop conditions

again, i.e. the counts are decremented before the new values are propagated to determine if the

process is at the end of the loop, and if it is, the TOP register 50 is then propagated to start the

next loop; the 'propagating' is not clearly defined and can be broadly interpreted to mean

propagating any of conditions at any time in the process).

23.   Referring to claim 23 Atkins has taught wherein the set of registers are a second set of

registers, the apparatus further including a first set of registers coupled to the control unit (Atkins

figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers; the registers are coupled to the control unit indirectly from the execution units, since the registers are a part of the execution units and the execution units are coupled to the control unit), wherein the control unit is further adapted to:

write the loop conditions of the hardware loop to the first set of registers prior to propagating at least one of the loop conditions to the second set of registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers).

24.   Referring to claim 24 Atkins has taught wherein at least one of the pipelines is a data address generation pipeline (Atkins column 6 lines 33-48, figure 3 reference D-BUS; at least a part of the pipeline is concerned with data address generation).

25.   Referring to claim 25 Atkins has taught wherein at least one of the pipelines is a system pipeline (Atkins figures 2-3, column 4 lines 13-44; all of the pipelines are a part of the system).

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

26.   Claim 19 is rejected under 35 U.S.C. 103(a) as being unpatentable over Atkins in view of Tran U.S. Patent Number 6,003,128 (herein referred to as Tran).

27.   Referring to claim 19 Atkins has not taught wherein the first set of registers are speculative registers. Tran has taught wherein the first set of registers are speculative registers (Tran abstract). Tran has taught that using speculative processing with predictions reduces the processing time needed by the system to complete a program (Tran column 2 line 34-column 3 line 5). One of ordinary skill in the art at the time of the invention would have recognized that adding speculative loop prediction and execution to Atkins would increase the speed of the loop execution taking place in the system of Atkins. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to implement speculative loop processing to increase the speed of execution of the instructions.

28.   Claims 26-30 are rejected under 35 U.S.C. 103(a) as being unpatentable over Atkins.

29.   Referring to claim 26 Atkins has taught a system comprising:

wherein a processor includes a first set of registers (Atkins figures 2 and 4; fig. 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers), a first pipeline, a second pipeline, and a control unit (Atkins figure 2 reference 26, column 4 lines 12-44) adapted to

first calculate a first loop condition of a first hardware loop from a loop setup instruction using a first arithmetic logic unit in the first pipeline (Atkins figures 2-4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; figure 2 and figure 3 reference 20-4C), second calculate a

second loop condition of said first hardware loop from the loop setup instruction using a second

arithmetic logic unit in the second pipeline (Atkins figures 2-4, column 2 lines 14-50, column 9

line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the

execution units, 20A-C; figure 2 and figure 3 reference 20-4C) , in parallel with the first

calculate (Atkins column 4 lines 12-44, column 3 lines 1-20; Atkins has several pipelines, where

several different loops can be executed in parallel with each other, since more than one

SETLOOP instruction could be present and each one can be present in a different pipeline; also

the situation would exist where the two pipelines are executing different instances of the same

instruction group, or same SETLOOP instruction, thereby being the same hardware loop e) via a

second pipeline of the pipelined processor); and

write the first and second loop conditions of said first hardware loop to the first set of

registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column

10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units,

20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3

setloop hardware registers).

30.   Atkins has not taught a static random access memory device and a processor coupled to

the static random access memory device. Atkins has taught a cache and a processor coupled to

the cache (Atkins figure 1). Official Notice is taken that it is well known in the art that caches are

implemented using static random access memory. One of ordinary skill in the art at the time of

the rejection would recognize that using static random access memory would allow the cache

memory to provide the instructions and data to the processor faster that dynamic random access

memory, thus making static random access memory a more desirable memory for caches.

Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use static random access memory as the cache memory of Atkins to provide the instructions and data more quickly to the processor, which reduces the time required by execution.

31. Referring to claim 27 Atkins has taught including a third pipeline, the control unit further adapted to:

calculate a third loop condition of the hardware loop from the loop setup instruction using a third arithmetic logic unit in the third pipeline (Atkins figures 2-4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; figure 2 and figure 3 reference 20-4C); and

write the first, second and third loop conditions of the hardware loop to the first set of registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers).

32. Referring to claim 28 Atkins has taught a system comprising:

wherein the processor includes a first set of registers (Atkins figures 2 and 4; fig. 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers), a second set of registers (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers

for the system), a first pipeline, a second pipeline, and a control unit (Atkins figure 2 reference 26, column 4 lines 12-44) adapted to:

write loop conditions of a first hardware loop to the first set of registers (Atkins figure 2 reference 20, figure 4, column 2 lines 14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the execution units, 20A-C; all of the registers used for the setloop hardware in all of the processors is a set of 3 setloop hardware registers);

propagate at least one of the loop conditions of said first hardware loop to the second set of registers via the first pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers for the system, the value of the TOP register is propagated to the instruction-address register set in all of the pipelines when necessary by the hardware performing a loop operation); and

propagate at least one of the loop conditions of said first hardware loop to the second set of registers via the second pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers for the system, the value of the TOP register is propagated to the instruction-address register set in all of the pipelines when necessary by the hardware performing a loop operation; also the situation would exist where the two pipelines are executing different instances of the same instruction group, or same SETLOOP instruction, thereby being the same hardware loop e) via a second pipeline of the pipelined processor); and

begin calculating data using said first hardware loop prior to completing said first and second propagate (Atkins figure 4; the registers included in the loop hardware shown in figure 4,

are used to calculate the current parameters for the loop, before propagating the loop conditions again, i.e. the counts are decremented before the new values are propagated to determine if the process is at the end of the loop, and if it is, the TOP register 50 is then propagated to start the next loop; the 'propagating' is not clearly defined and can be broadly interpreted to mean propagating any of conditions at any time in the process).

33.   Atkins has not taught a static random access memory device and a processor coupled to the static random access memory device. Atkins has taught a cache and a processor coupled to the cache (Atkins figure 1). Official Notice is taken that it is well known in the art that caches are implemented using static random access memory. One of ordinary skill in the art at the time of the rejection would recognize that using static random access memory would allow the cache memory to provide the instructions and data to the processor faster that dynamic random access memory, thus making static random access memory a more desirable memory for caches. Therefore, it would have been obvious to one of ordinary skill in the art at the time of the invention to use static random access memory as the cache memory of Atkins to provide the instructions and data more quickly to the processor, which reduces the time required by execution.

34.   Referring to claim 29 Atkins has taught further including a third pipeline, the control unit further adapted to propagate at least one of the loop conditions to the second set of registers via the third pipeline (Atkins figure 4 reference 50, column 10 lines 16-24; the instruction-address registers for all of the execution units together are a set of 3 instruction-address registers for the system, the value of the TOP register is propagated to the instruction-address register set in all of the pipelines when necessary by the hardware performing a loop operation).

35.   Referring to claim 30 Atkins has taught further adapted to:

calculate a first loop condition of the hardware loop from a loop setup instruction using a

first arithmetic logic unit in the first pipeline (Atkins figures 2-4, column 2 lines 14-50, column 9

line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is present in all of the

execution units, 20A-C); and

calculate a second loop condition of the hardware loop from the loop setup instruction

using a second arithmetic logic unit in the second pipeline (Atkins figures 2-4, column 2 lines

14-50, column 9 line 66-column 10 line 55; figure 4 is the hardware for the setloop, which is

present in all of the execution units, 20A-C).

Response to Arguments

36.   Applicant's arguments filed 09/02/04 have been fully considered but they are not

persuasive.

37.   In the remarks, in regard to the rejections of the claims, Applicant argues in essence that:

> "...Atkins, who does not teach or suggest multiple parallel pipelines propagating
> information for the same hardware loop conditions. Therefore, claim 1 as amended should
> obviate the rejection thereto, and therefore claim I should be allowable along with the
> claims which depend therefrom."

38.   This is not found persuasive. Although the amendment by applicant has limited the

claims to having a first and second pipeline, in parallel, propagating loop conditions for the same

hardware loop, the amendment does not overcome the situation where the two pipelines are

executing different instances of the same instruction group, or same SETLOOP instruction,

thereby being the same hardware loop. Also, Applicant has assumed that by stating that the loop

conditions are propagating in different pipelines "in parallel", that it is analogous with being

concurrent. In addition, the specification does not give a clear definition of a "hardware loop",
since on page one of the specification a "hardware loop" is described as being, "a "hardware
loop," which may be dedicated hardware designed to expedite the execution of software
instructions within a loop construct", and includes the words "may be".

Conclusion

Any inquiry concerning this communication or earlier communications from the
examiner should be directed to Charles A Harkness whose telephone number is 571-272-4167.
The examiner can normally be reached on 8Flex.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's
supervisor, Eddie Chan can be reached on 571-272-4162. The fax phone number for the
organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent
Application Information Retrieval (PAIR) system. Status information for published applications
may be obtained from either Private PAIR or Public PAIR. Status information for unpublished
applications is available through Private PAIR only. For more information about the PAIR
system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR
system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Charles Allen Harkness

Patent Examiner

Art Unit 2183

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100

November 11, 2004